DW03_bictr_decode

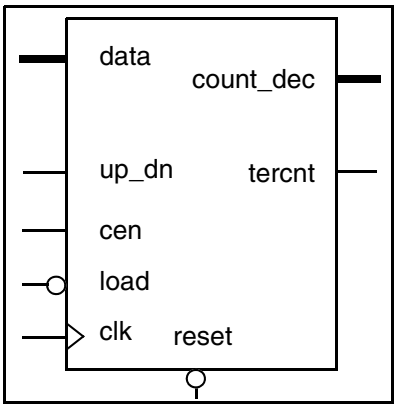
Up/Down Binary Counter with Output Decode

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Up/down count control
- Asynchronous reset
- Loadable count register
- Counter enable
- Terminal count flag

Revision History



Description

DW03_bictr_decode is a general-purpose up/down counter whose outputs are binary decoded.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
data	*width*	Input	Counter load input
up_dn	1	Input	High for count up and low for count down
load	1	Input	Enable data load to counter, active low
cen	1	Input	Count enable, active high
clk	1	Input	Clock
reset	1	Input	Counter reset, active low
count_dec	2^{width}	Output	Binary decoded count value
tercnt	1	Output	Terminal count flag

Table 1-2 Parameter Description

Parameter	Values	Function
width	≥ 1	Width of data input bus

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW03.DW03_BICTR_DECODE_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_bictr_decode_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_bictr_decode.v	Verilog simulation model source code

Table 1-5 Counter Operation Truth Table

reset	load	cen	up_dn	Operation
0	X	X	X	Reset
1	0	X	X	Load
1	1	0	X	Standby
1	1	1	0	Count down
1	1	1	1	Count up

The counter is *width* bits wide, and has 2^{width} states from “000...0” to “111...1”. The counter is clocked on the positive edge of `clk`.

The `reset`, active low, provides for an asynchronous reset of the counter to “000...0”. If the `reset` pin is connected to ‘1’, then the reset logic is not synthesized, resulting in a smaller and faster counter.

The `up_dn` input controls whether the counter counts up (`up_dn` high) or down (`up_dn` low), starting on the next positive edge of `clk`.

The counter is loaded with `data` by asserting `load` (low) and applying data to `data`. The data load operation is synchronous with respect to the positive edge of `clk`.

When the count enable pin, `cen`, is high, the counter is active. When `cen` is low, the counter is disabled, and `count` remains at the same value.

The `count_dec` is an output bus that ranges from $2^{width-1}$ to 0.

The `tercnt` is an output flag that is asserted one state before the count rolls over. When counting up, `tercnt` is high at `count_dec` = “100...000”. When counting down, `tercnt` is high at `count_dec` = “000.....001”.

Timing Diagrams

Figure 1-1 through Figure 1-4 on page 4 show various timing diagrams for DW03_bictr_decode.

Figure 1-1 Functional Operation: Reset, Load, and Count Enable

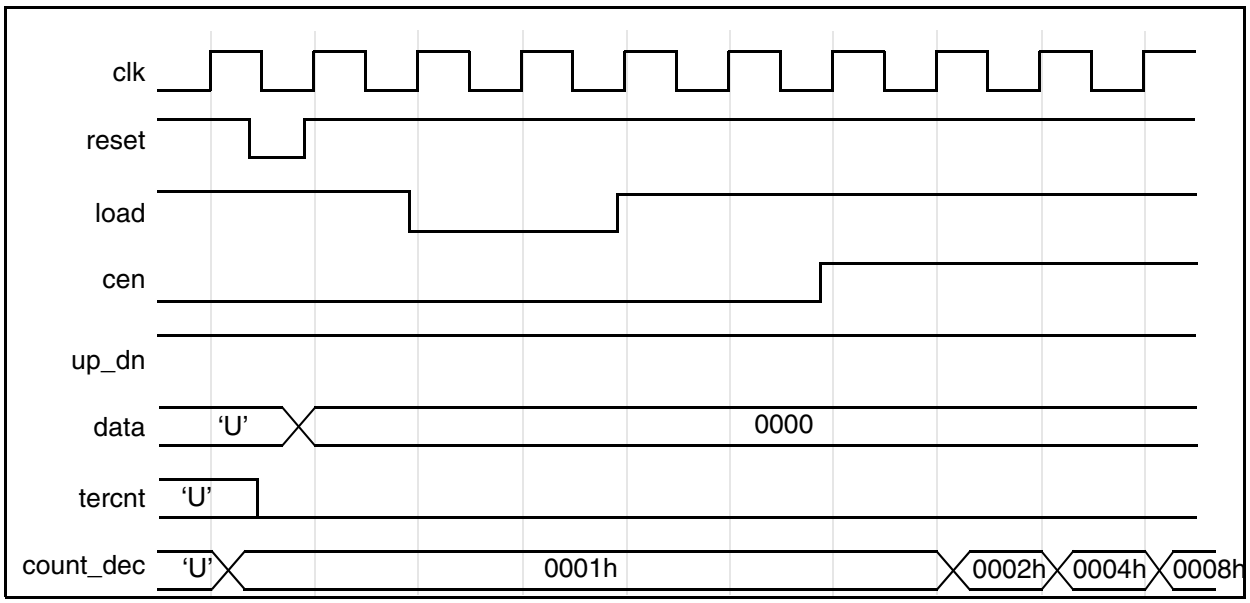


Figure 1-2 Functional Operation: Up and Down Counting

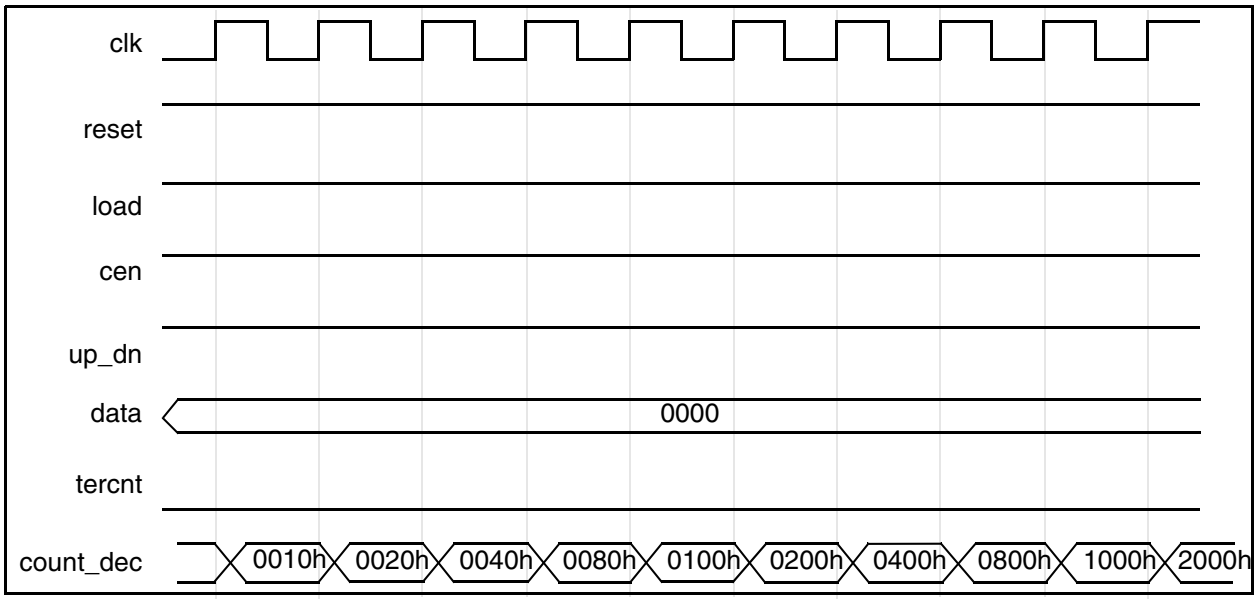


Figure 1-3 Functional Operation: Terminal Count Flag

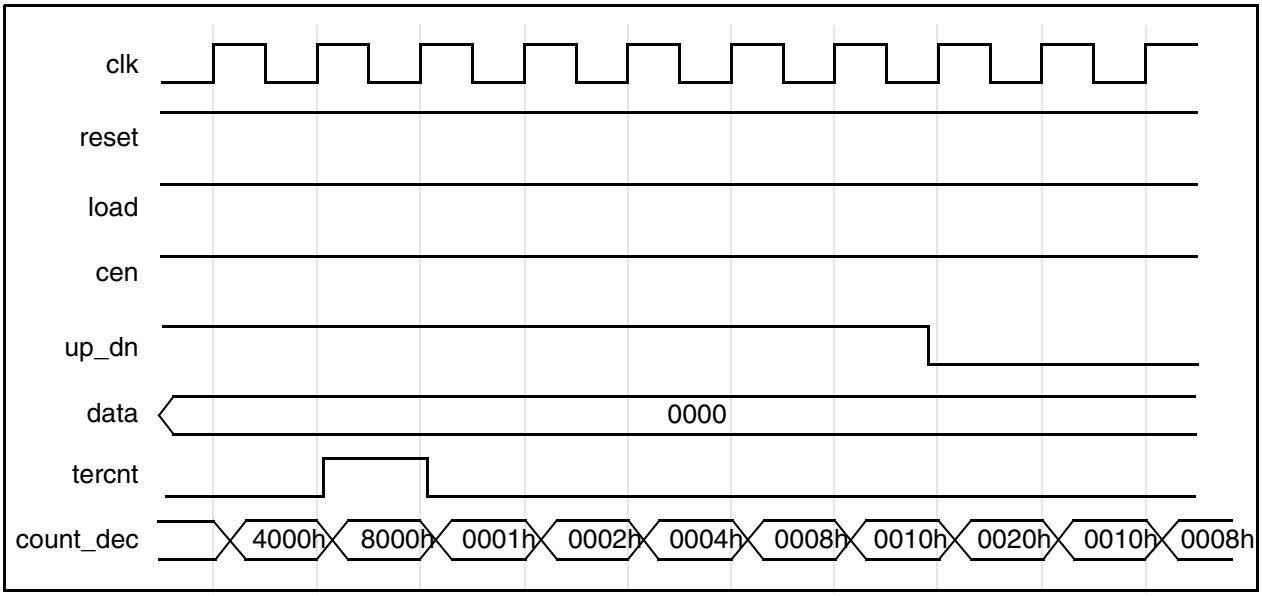
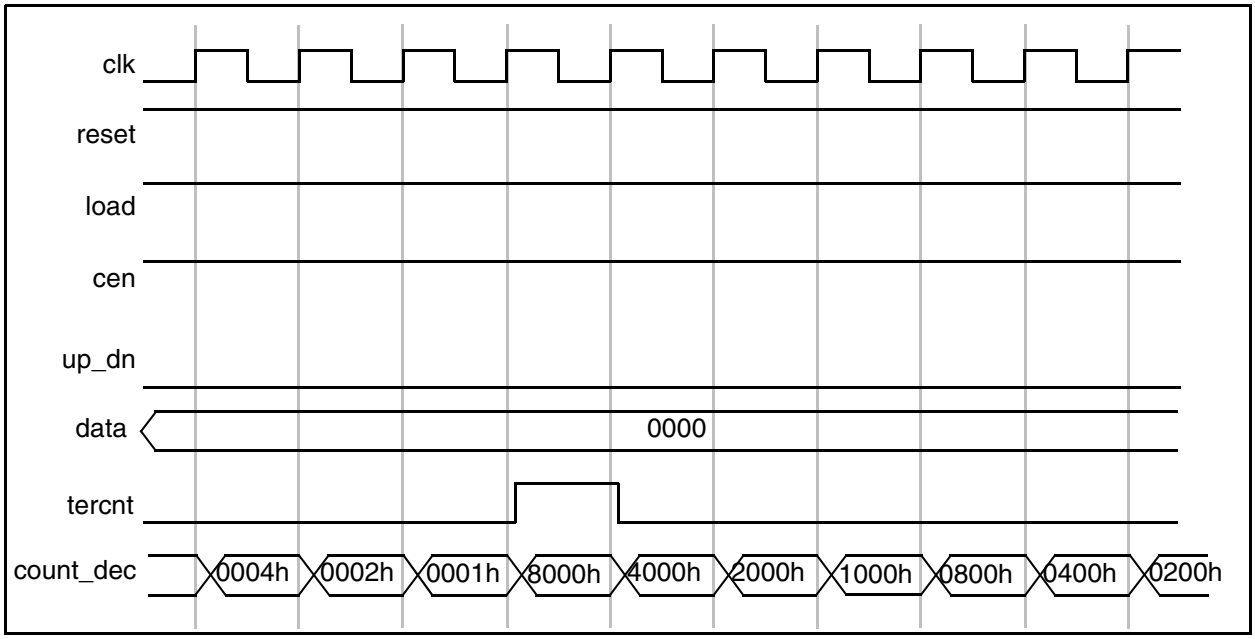


Figure 1-4 Functional Operation: Effect on Terminal Count



Related Topics

- Logic – Sequential Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW03_bictr_decode_inst is
  generic (inst_width : POSITIVE := 8);
  port (inst_data      : in std_logic_vector(inst_width-1 downto 0);
        inst_up_dn     : in std_logic;
        inst_load      : in std_logic;
        inst_cen       : in std_logic;
        inst_clk       : in std_logic;
        inst_reset     : in std_logic;
        count_dec_inst : out std_logic_vector(2**inst_width-1 downto 0);
        tercnt_inst    : out std_logic);
end DW03_bictr_decode_inst;

architecture inst of DW03_bictr_decode_inst is
begin

  -- Instance of DW03_bictr_decode
  U1 : DW03_bictr_decode
    generic map ( width => inst_width )
    port map ( data => inst_data, up_dn => inst_up_dn,
               load => inst_load, cen => inst_cen, clk => inst_clk,
               reset => inst_reset, count_dec => count_dec_inst,
               tercnt => tercnt_inst );
end inst;

-- pragma translate_off
configuration DW03_bictr_decode_inst_cfg_inst of DW03_bictr_decode_inst is
  for inst
  end for; -- inst
end DW03_bictr_decode_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW03_bictr_decode_inst( inst_data, inst_up_dn,
                               inst_load, inst_cen, inst_clk,
                               inst_reset, count_dec_inst, tercnt_inst );

  parameter width = 8;

  input [width-1 : 0] inst_data;
  input inst_up_dn;
  input inst_load;
  input inst_cen;
  input inst_clk;
  input inst_reset;
  output [(1<<width)-1 : 0] count_dec_inst;
  output tercnt_inst;

  // Instance of DW03_bictr_decode
    DW03_bictr_decode #(width)
      U1 ( .data(inst_data), .up_dn(inst_up_dn), .load(inst_load),
           .cen(inst_cen), .clk(inst_clk), .reset(inst_reset),
           .count_dec(count_dec_inst), .tercnt(tercnt_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
March 2019	DWBB_201903.0	■ Removed minPower designation from this datasheet
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com